Fig.1

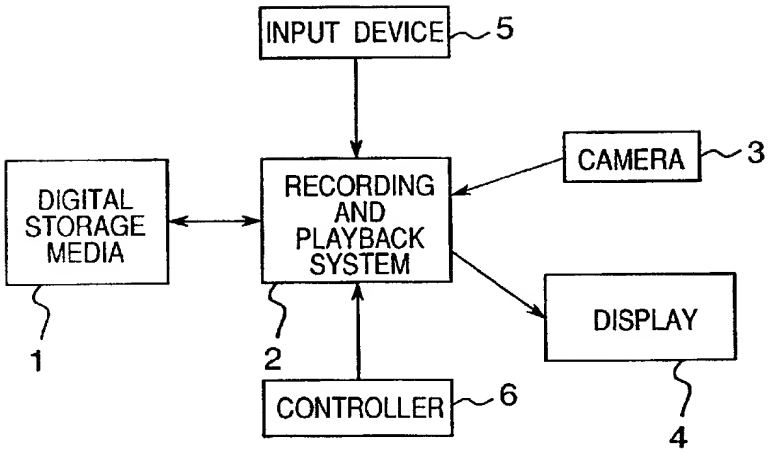

INPUT DEVICE
5
CAMERA
3
DIGITAL STORAGE MEDIA
RECORDING AND PLAYBACK SYSTEM
DISPLAY
1
2
CONTROLLER
6
4

Fig.2

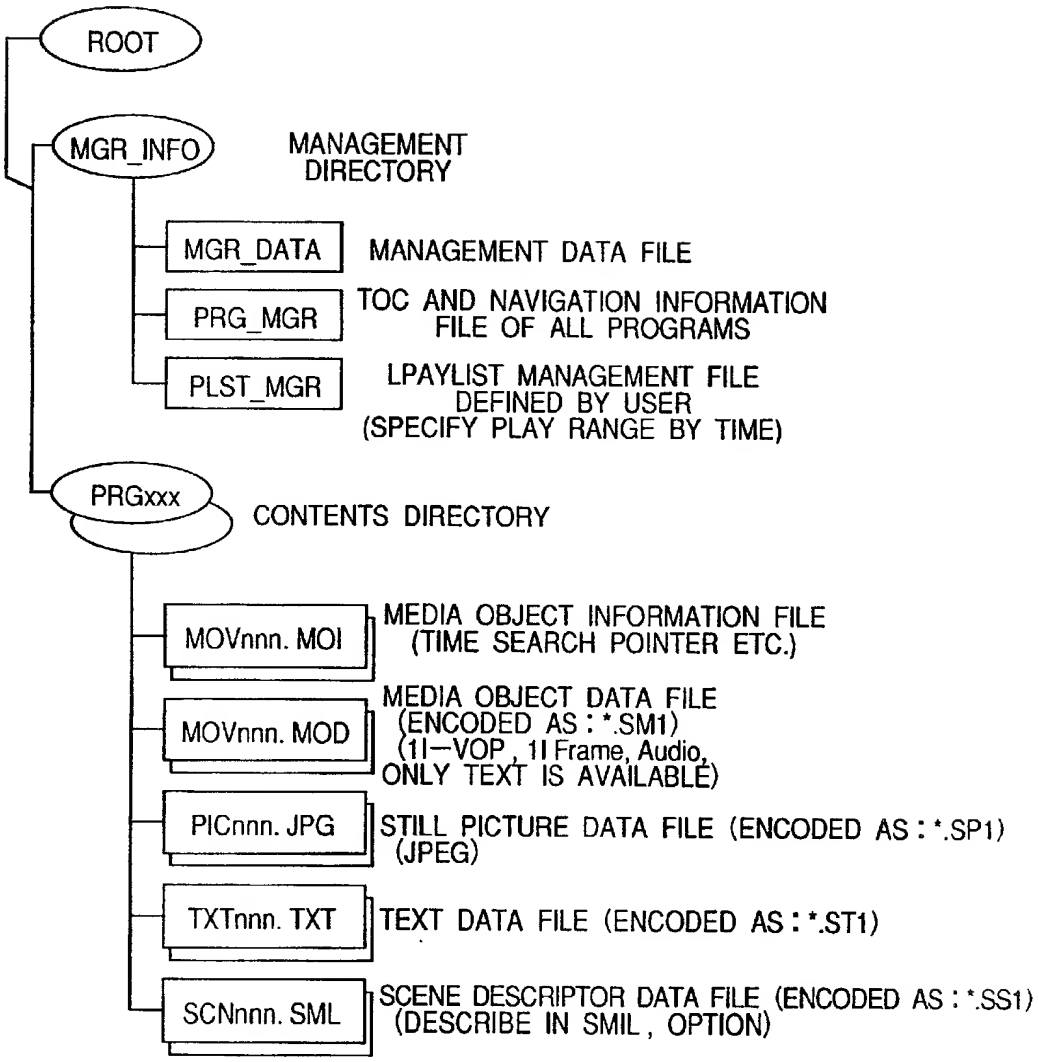

ROOT
MGR_INFO
MANAGEMENT DIRECTORY
MGR_DATA
MANAGEMENT DATA FILE
PRG_MGR
TOC AND NAVIGATION INFORMATION FILE OF ALL PROGRAMS
PLST_MGR
LPAYLIST MANAGEMENT FILE DEFINED BY USER (SPECIFY PLAY RANGE BY TIME)
PRGxxx
CONTENTS DIRECTORY
MOVnnn. MOI
MEDIA OBJECT INFORMATION FILE (TIME SEARCH POINTER ETC.)
MOVnnn. MOD
MEDIA OBJECT DATA FILE (ENCODED AS : *.SM1) (1I—VOP , 1I Frame, Audio, ONLY TEXT IS AVAILABLE)
PICnnn. JPG
STILL PICTURE DATA FILE (ENCODED AS : *.SP1) (JPEG)
TXTnnn. TXT
TEXT DATA FILE (ENCODED AS : *.ST1)
SCNnnn. SML
SCENE DESCRIPTOR DATA FILE (ENCODED AS : *.SS1) (DESCRIBE IN SMIL , OPTION)

Fig.3

MANAGEMENT DATA FILE (MGR_DATA)

FIELD NAME	CONTENT	SIZE (bit)
USHORT DataType	MGR_DATA TYPE (FIXED)	16
USHORT DataSize	MGR_DATA SIZE	16
USHORT Version	VERSION	16
OBJPOSITION ResumeMarker	PROGRAM ID + OFFSET (ms)	32+32
BYTE TextInfo [200]	TEXT INFORMATION	200 Bytes

31

Fig.4

PROGRAM MANAGER FILE (PRG_MGR)

FIELD NAME	CONTENT	SIZE (bit)
USHORT DataType	PRG_MGR TYPE (FIXED)	16
USHORT DataSize	PRG_MGR SIZE	16
ULONG PlayBackDuration	PLAYBACK DURATION OF ALL PROGRAM (ms)	32
USHORT NumPrgInfo	NUMBER OF PROGRAM INFORMATION	16
PRG_INFO PrgInfoTbl [NumPrgInfo]	TABLE OF PROGRAM INFORMATION	Variable

Fig.5

PROGRAM INFORMATION (PRG_INFO)

FIELD NAME	CONTENT	SIZE (bit)
USHORT DataType	PRG_INFO TYPE (FIXED)	16
USHORT DataSize	PRG_INFO SIZE	16
51 OBJECTID PrgID	PROGRAM ID	32
52 ULONG PlaybackDuration	PLAYBACK DURATION (ms)	32
USHORT Attribute	ATTRIBUTE (USE PROTECT, SCENE DESCRIPTION ?)	16
USHORT Profile	PROFILE INFORMATION	16
BYTE TextInfo [200]	TEXT INFORMATION (TITLE)	204 Byte
ULONG RepPos	SPECIFY THE PLACE WHERE MAIN PICTURE EXIST	64
53 USHORT NumRefMoi	NUMBER OF MEDIA OBJECT MANAGED BY THIS PROGRAM	16
54 ULONG RefMoiTbl [NumRefMoi]	OBJECT ID OF MEDIA OBJECT, PLAYBACK DURATION, TABLE SHOWING PRESENCE AND ABSENCE SUBORDINATION OBJECT	64*NumObjIDTbl
		8
BYTE NumMarker	NUMBER OF MARKER INFORMATION	8
ULONG MarkerTbl [NumMarker]	MARKER INFORMATION (OFFSET VALUE ms) TABLE	32*NumMarker

PRG_INFO2

PRG_INFO1

Fig.6

PLAY LIST MANAGER FILE (PLST_MGR)

FIELD NAME	CONTENT	SIZE (bit)
USHORT DataType	PLST_MGR TYPE (FIXED)	16
USHORT DataSize	PLST_MGR SIZE	16
USHORT NumPlstInfo	NUMBER OF PLAYLIST INFORMATION	16
PLST_INFO PlstInfoTbl [NumPlstInfo]	TABLE OF PLAYLIST INFORMATION	Variable

Fig.7

PLAYLIST INFORMATION (PLST_INFO)

FIELD NAME			CONTENT	SIZE (bit)	
USHORT DataType			PRG_INFO TYPE (FIXED)	16	
USHORT DataSize			PRG_INFO SIZE	16	
ULONG PlayBackDuration			PLAYBACK DURATION (ms)	32	
ULONG Attribute			ATTRIBUTE (PROTECT)	16	
BYTE TextInfo [200]			TEXT INFORMATION (TITLE)	200 Byte	
ULONG RepPos			SPECIFY THE PLACE WHERE MAIN PICTURE EXIST	64	
USHORT NumPrgID			NUMBER OF PLAYBACK PROGRAM INFORMATION MANAGED BY THIS PLAYLIST	16	
71	PLAYBACK PROGRAM INFORMATION	ULONG ObjID	OBJECT ID OF PROGRAM	32	X NumPrgID
		ULONG StartPos	PLAYBACK START TIME (ms)	32	
		ULONG EndPos	PLAYBACK END TIME (ms)	32	
BYTE NumMarker			NUMBER OF MARKER INFORMATION	8	
ULONG MarkerTbl [NumMarker]			MARKER INFORMATION (OFFSET VALUE ms) TABLE	32*NumMarker	

Fig.8

MEDIA OBJECT INFORMATION FILE (*.MOI)

	FIELD NAME			CONTENT	SIZE (bit)	
	USHORT DataType			MOI TYPE (FIXED)	16	
	USHORT DataSize			MOI SIZE	16	
82	Playback Duration			MOI PLAYBACK DURATION PTm	4	
	ATTRIBUTE TextAttr			TEXT CODE USED FOR TEXT DATA ETC.	128	
	BYTE TstType			TIME SEARCH TABLE TYPE (Tst Type=1,2,3)	8	
TstType=1	USHORT TstInterval			RESOLVING POWER OF TIME SEARCH TABLE (ms)	16	
	USHORT FrameTime			REPRESENT 1 FRAME TIME WITH FRACTION (ms)	32	
	USHORT NumTstEntry			TIME SEARCH TABLE ENTRY NUMBER	16	
	UINT16NumModui			MODU INFORMATION TABLE NUMBER	16	
	MODU INFO ModuiTbl [NUmModui]			MODU INFORMATION TABLE	48*NumModui	
81	ENTRY	UINT16 ModuNumber		MODU NUMBER	16	XNumTstEntry1
		UINT8 EntryFrameDi		FRAME NUMBER FROM ONE PREVIOUS ENTRY FRAME TO TIME SEARCH ENTRY	8	
		UINT32 ModuOffset		MODU POSITION (byte)	32	
TstType=2	USHORT TstInterval			RESOLVING POWER OF TIME SEARCH TABLE (ms)	16	
	USHORT NumTstEntry2			TIME SEARCH TABLE ENTRY NUMBER	16	
	ENTRY	UINT8 EntryFrameDiff		FRAME NUMBER FROM ONE PREVIOUS ENTRY FRAME TO TIME SEARCH ENTRY	8	X NumTstEntry2
		UINT32 ModuOffset		MODU POSITION (byte)	32	
TstType=3	UINT32 FrameTime			REPRESENT 1 FRAME PLAYBACK TIME WITH FRACTION (ms)	32	
	ULONG PacketSize			PACKET SIZE (BYTE)	32	
	BYTE NumFrame			FRAME NUMBER IN 1 PACKET	8	

Fig.9

MEDIA OBJECT UNIT INFORMATION (MODU_INFO)

FIELD NAME	CONTENT	SIZE (bit)
USHORT EntrySize	Entry Frame SIZE (Byte)	20
USHORT ModuPbTime	FRAME NUMBER CONSTRUCT MODU	6
USHORT ModuSize	MODU SIZE (byte)	22

Fig.10

EXAMPLE OF RECORDING PROCESS
DETECT NEW RECORDING MEDIA
  CREATE SD-VIDEO DIRECTORY, MGP_INFO DIRECTORY, PRG_MGR FILE AND DETECT FORMATTING RECORD MODE ON
  CHECK PROGRAM NAMES IN Root DIRECTRY TO DETERMINE NEXT PROGRAM NUMBER nnn
   (TYPICALLY HIGHEST EXISTING PROGRAM NUMBER + 1, IF ALREADY MAXIMUM THEN NEXT AVAILABLE NUMBER,
    IF NO NUMBER IS AVAILABLE THEN WARNING IS ISSUED.)
    CREATE PROGRAM DIRECTORY (PRGnnn)
ABOVE-PROCEDURE IS THEREAFTER REPEATED

DETECT FIRST Rec OPERATION
  START RECORDING OF MEDIA OBJECT DATA FILE (MOV001.MOD)
  CREATE MEDIA OBJECT INFORMATION FILE (MOV001.MOI)

DETECT Stop OPERATION
  END MOD FILE RECORDING, UPDATE MEDIA OBJECT INFORMATION FILE (MOV001.MOI)
  (dataSize, NumModui, ModuiTbl, NumTstEntry1, TstEntry1)
  CREATE SCENE DESCRIPTOR DATA FILE (SCN001.SML)
  UPDATE PRG_INFO (DataSize, PlaybackDuration, NumRefMoi, RefmoiTbl)

DETECT RECORDING MODE OFF
  UPDATE PROGRAM MANAGER (PRG_MGR) (DataSize, PlaybackDuration, NumPrgInfo)

Fig.11

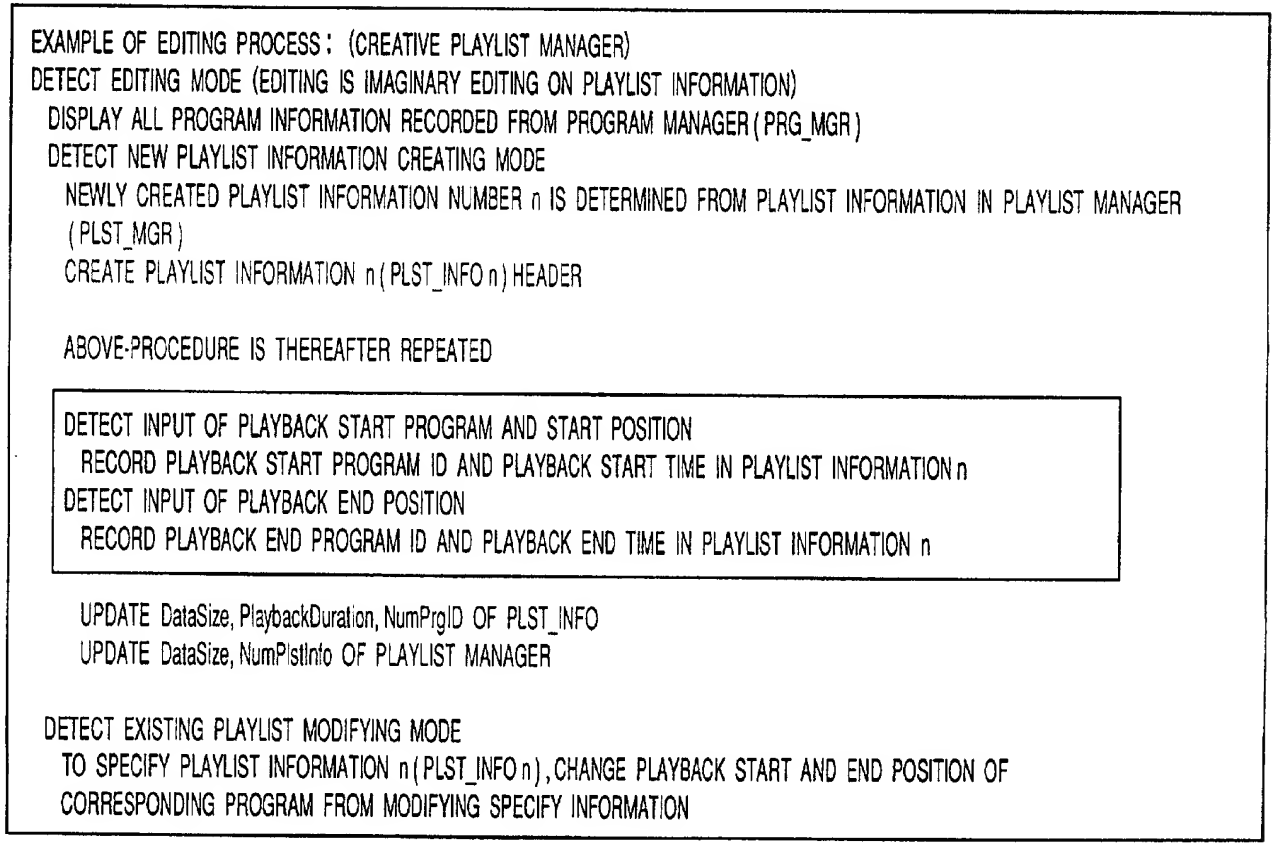

EXAMPLE OF EDITING PROCESS : (CREATIVE PLAYLIST MANAGER)
DETECT EDITING MODE (EDITING IS IMAGINARY EDITING ON PLAYLIST INFORMATION)
DISPLAY ALL PROGRAM INFORMATION RECORDED FROM PROGRAM MANAGER (PRG_MGR)
DETECT NEW PLAYLIST INFORMATION CREATING MODE
NEWLY CREATED PLAYLIST INFORMATION NUMBER n IS DETERMINED FROM PLAYLIST INFORMATION IN PLAYLIST MANAGER (PLST_MGR)
CREATE PLAYLIST INFORMATION n (PLST_INFO n) HEADER
ABOVE-PROCEDURE IS THEREAFTER REPEATED
DETECT INPUT OF PLAYBACK START PROGRAM AND START POSITION
RECORD PLAYBACK START PROGRAM ID AND PLAYBACK START TIME IN PLAYLIST INFORMATION n
DETECT INPUT OF PLAYBACK END POSITION
RECORD PLAYBACK END PROGRAM ID AND PLAYBACK END TIME IN PLAYLIST INFORMATION n
UPDATE DataSize, PlaybackDuration, NumPrgID OF PLST_INFO
UPDATE DataSize, NumPlstInfo OF PLAYLIST MANAGER
DETECT EXISTING PLAYLIST MODIFYING MODE
TO SPECIFY PLAYLIST INFORMATION n (PLST_INFO n), CHANGE PLAYBACK START AND END POSITION OF CORRESPONDING PROGRAM FROM MODIFYING SPECIFY INFORMATION

Fig.12

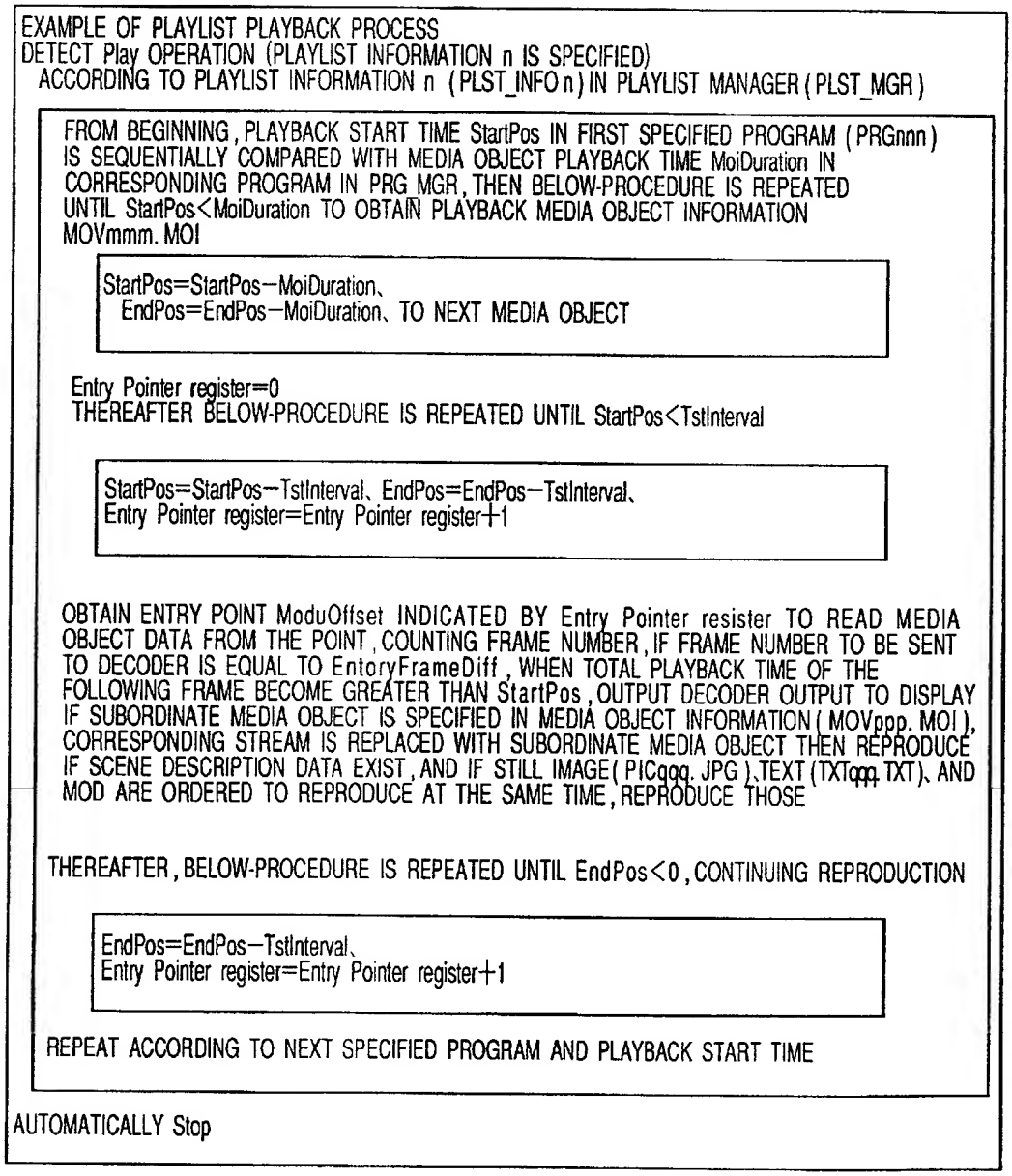

EXAMPLE OF PLAYLIST PLAYBACK PROCESS
DETECT Play OPERATION (PLAYLIST INFORMATION n IS SPECIFIED)
ACCORDING TO PLAYLIST INFORMATION n (PLST_INFO n) IN PLAYLIST MANAGER (PLST_MGR)
FROM BEGINNING, PLAYBACK START TIME StartPos IN FIRST SPECIFIED PROGRAM (PRGnnn) IS SEQUENTIALLY COMPARED WITH MEDIA OBJECT PLAYBACK TIME MoiDuration IN CORRESPONDING PROGRAM IN PRG MGR, THEN BELOW-PROCEDURE IS REPEATED UNTIL StartPos<MoiDuration TO OBTAIN PLAYBACK MEDIA OBJECT INFORMATION MOVmmm.MOI
StartPos=StartPos−MoiDuration、
EndPos=EndPos−MoiDuration、TO NEXT MEDIA OBJECT
Entry Pointer register=0
THEREAFTER BELOW-PROCEDURE IS REPEATED UNTIL StartPos<TstInterval
StartPos=StartPos−TstInterval、EndPos=EndPos−TstInterval、
Entry Pointer register=Entry Pointer register+1
OBTAIN ENTRY POINT ModuOffset INDICATED BY Entry Pointer resister TO READ MEDIA OBJECT DATA FROM THE POINT, COUNTING FRAME NUMBER, IF FRAME NUMBER TO BE SENT TO DECODER IS EQUAL TO EntoryFrameDiff, WHEN TOTAL PLAYBACK TIME OF THE FOLLOWING FRAME BECOME GREATER THAN StartPos, OUTPUT DECODER OUTPUT TO DISPLAY IF SUBORDINATE MEDIA OBJECT IS SPECIFIED IN MEDIA OBJECT INFORMATION (MOVppp.MOI), CORRESPONDING STREAM IS REPLACED WITH SUBORDINATE MEDIA OBJECT THEN REPRODUCE IF SCENE DESCRIPTION DATA EXIST, AND IF STILL IMAGE (PICqqq.JPG)、TEXT (TXTqqq.TXT)、AND MOD ARE ORDERED TO REPRODUCE AT THE SAME TIME, REPRODUCE THOSE
THEREAFTER, BELOW-PROCEDURE IS REPEATED UNTIL EndPos<0, CONTINUING REPRODUCTION
EndPos=EndPos−TstInterval、
Entry Pointer register=Entry Pointer register+1
REPEAT ACCORDING TO NEXT SPECIFIED PROGRAM AND PLAYBACK START TIME
AUTOMATICALLY Stop

Fig.13

EXAMPLE OF RANDOM PLAYBACK PROCEDURE

USER SPECIFY PROGRAM nnn AND START / END TIME ON TOC DISPLAY CREATED

FROM PROGRAM MANAGER (PRG_MGR)

DETECT Play OPERATION

CHECK MEDIA OBJECT INFORMATION (MOVmmm. MOI) IN SPECIFIED PROGRAM (PRGnnn) IN SEQUENCE FROM BEGINNING, AND SUBTRACTING PLAYBACK TIME (PlayDuration) FROM USER-SPECIFIED START TIME, DETECT MEDIA OBJECT NUMBER ppp INCLUDES START TIME.

FROM MOI ENTRY POINT TIME RESOLVING ABILITY (TstInterval), DETECT PLAYBACK START Packet POSITION AND Ipacket POSITION, AND THEN START REPRODUCTION AT THE MEDIA OBJECT DATA (MOVppp. MOD) SPECIFY POSITION

IF SUBORDINATION MEDIA OBJECT INFORMATION IS SPECIFIED IN MEDIA OBJECT INFORMATION (MOVppp. MOI), CORRESPONDING STREAM IS REPLACED WITH SUBORDINATION AND THEN REPRODUCE

IF SCENE DESCRIPTION DATA EXIST, AND IF STILL IMAGE (PICqqq. JPG), TEXT (TXTqqq. TXT) AND MOD ARE ORDERED TO BE REPRODUCED AT THE SAME TIME, REPRODUCE THOSE

THE ABOVE-PROCEDURE IS THEREAFTER REPEATED

NEXT MEDIA OBJECT DATA (MOVppp. MOD) IS REPRODUCED TOO

IF SUBORDINATION MEDIA OBJECT IS SPECIFIED IN MEDIA OBJECT INFORMATION (MOVppp. MOI), CORRESPONDING STREAM IS REPLACED WITH SUBORDINATION AND THEN REPRODUCE

IF SCENE DESCRIPTION DATA EXIST, AND IF STILL IMAGE (PICqqq. JPG), TEXT (TXTqqq. TXT) AND MOD ARE ORDERED TO BE REPRODUCED AT THE SAME TIME, REPRODUCE THOSE

Stop AT MOD PLAYBACK POINT OF PROGRAM WHICH ACCORD WITH END TIME

Fig.14

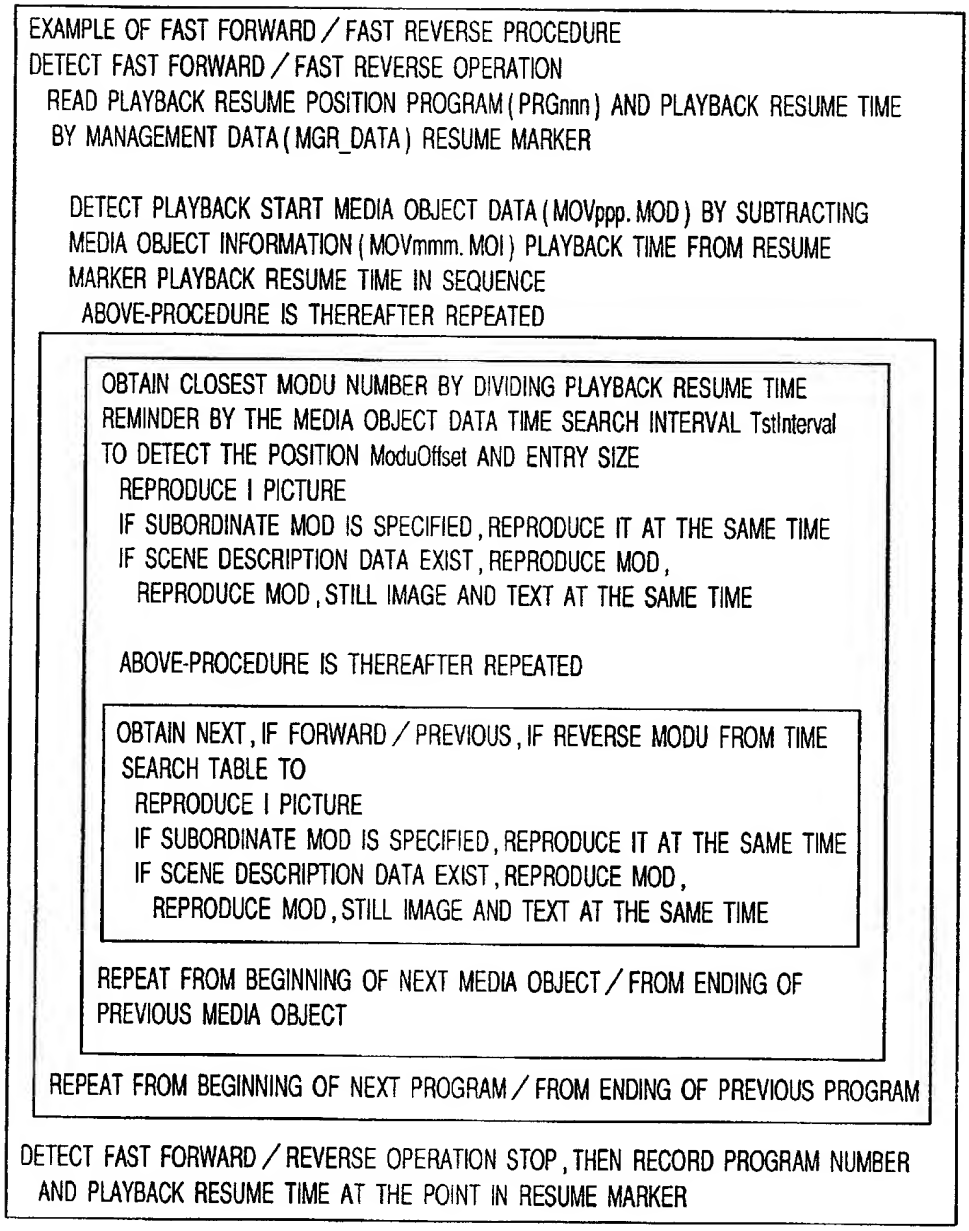

EXAMPLE OF FAST FORWARD / FAST REVERSE PROCEDURE
DETECT FAST FORWARD / FAST REVERSE OPERATION
READ PLAYBACK RESUME POSITION PROGRAM (PRGnnn) AND PLAYBACK RESUME TIME BY MANAGEMENT DATA (MGR_DATA) RESUME MARKER
DETECT PLAYBACK START MEDIA OBJECT DATA (MOVppp.MOD) BY SUBTRACTING MEDIA OBJECT INFORMATION (MOVmmm.MOI) PLAYBACK TIME FROM RESUME MARKER PLAYBACK RESUME TIME IN SEQUENCE
ABOVE-PROCEDURE IS THEREAFTER REPEATED
OBTAIN CLOSEST MODU NUMBER BY DIVIDING PLAYBACK RESUME TIME REMINDER BY THE MEDIA OBJECT DATA TIME SEARCH INTERVAL TstInterval TO DETECT THE POSITION ModuOffset AND ENTRY SIZE
REPRODUCE I PICTURE
IF SUBORDINATE MOD IS SPECIFIED, REPRODUCE IT AT THE SAME TIME
IF SCENE DESCRIPTION DATA EXIST, REPRODUCE MOD, REPRODUCE MOD, STILL IMAGE AND TEXT AT THE SAME TIME
ABOVE-PROCEDURE IS THEREAFTER REPEATED
OBTAIN NEXT, IF FORWARD / PREVIOUS, IF REVERSE MODU FROM TIME SEARCH TABLE TO
REPRODUCE I PICTURE
IF SUBORDINATE MOD IS SPECIFIED, REPRODUCE IT AT THE SAME TIME
IF SCENE DESCRIPTION DATA EXIST, REPRODUCE MOD, REPRODUCE MOD, STILL IMAGE AND TEXT AT THE SAME TIME
REPEAT FROM BEGINNING OF NEXT MEDIA OBJECT / FROM ENDING OF PREVIOUS MEDIA OBJECT
REPEAT FROM BEGINNING OF NEXT PROGRAM / FROM ENDING OF PREVIOUS PROGRAM
DETECT FAST FORWARD / REVERSE OPERATION STOP, THEN RECORD PROGRAM NUMBER AND PLAYBACK RESUME TIME AT THE POINT IN RESUME MARKER

Fig. 15

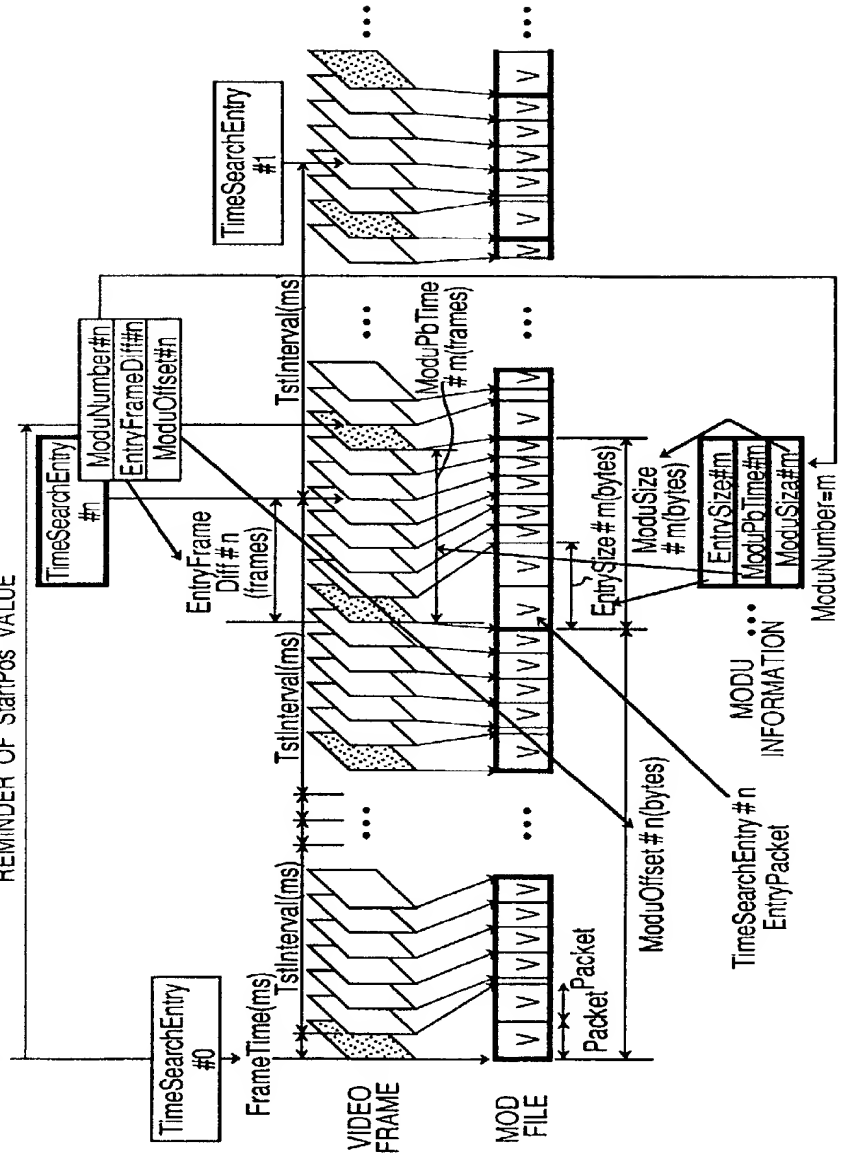

REMINDER OF StartPos VALUE
TimeSearchEntry #0
FrameTime(ms)
TstInterval(ms)
VIDEO FRAME
MOD FILE
Packet
Packet
TimeSearchEntry #n
ModuNumber#n
EntryFrameDiff#n
ModuOffset#n
EntryFrame Diff #n (frames)
TstInterval(ms)
TstInterval(ms)
TimeSearchEntry #1
ModuPbTime # m(frames)
EntrySize # m(bytes)
ModuSize # m(bytes)
ModuOffset # n(bytes)
TimeSearchEntry # n EntryPacket
MODU INFORMATION
EntrySize#m
ModuPbTime#m
ModuSize#m
ModuNumber=m

Fig.16

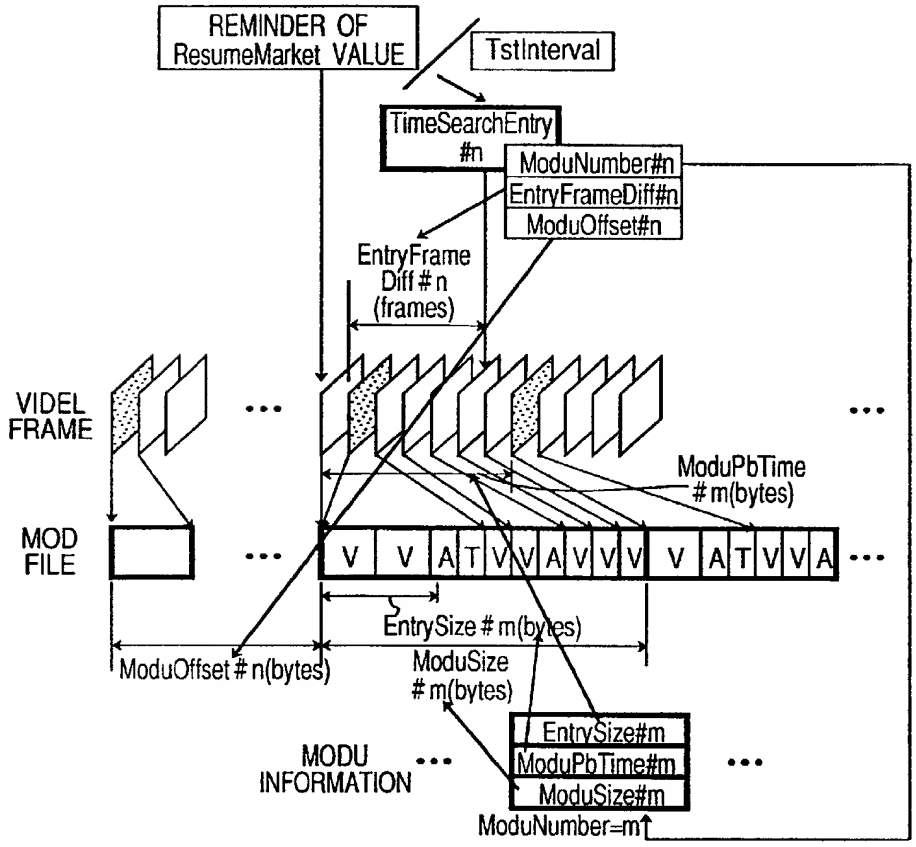

REMINDER OF ResumeMarket VALUE
TstInterval
TimeSearchEntry #n
ModuNumber#n
EntryFrameDiff#n
ModuOffset#n
EntryFrame Diff # n (frames)
VIDEL FRAME
MOD FILE
ModuPbTime # m(bytes)
V V A T V V A V V V V A T V V A
EntrySize # m(bytes)
ModuOffset # n(bytes)
ModuSize # m(bytes)
MODU INFORMATION
EntrySize#m
ModuPbTime#m
ModuSize#m
ModuNumber=m

Fig.17

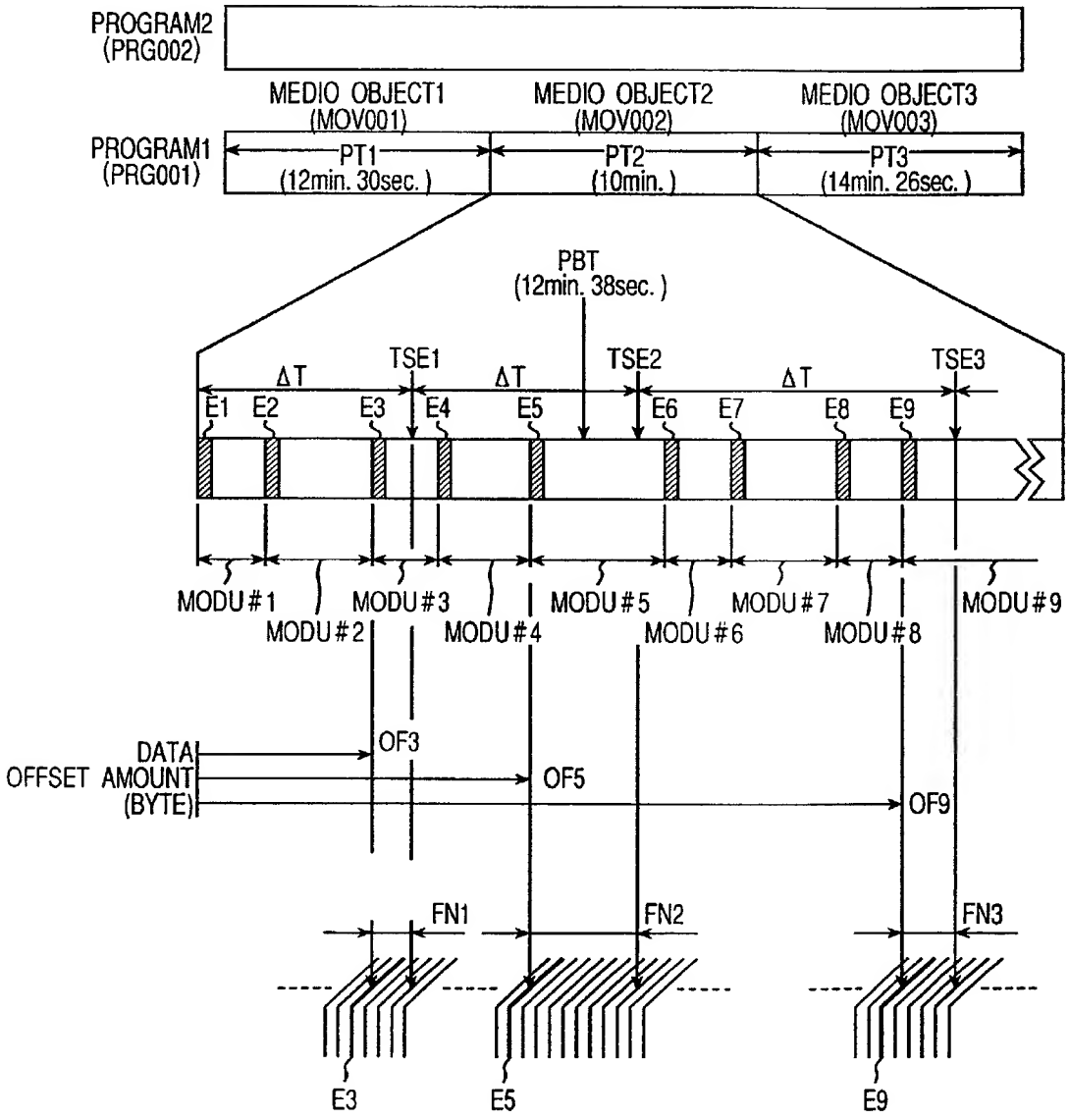

PROGRAM2
(PRG002)
MEDIO OBJECT1
(MOV001)
MEDIO OBJECT2
(MOV002)
MEDIO OBJECT3
(MOV003)
PROGRAM1
(PRG001)
PT1
(12min. 30sec.)
PT2
(10min.)
PT3
(14min. 26sec.)
PBT
(12min. 38sec.)
ΔT
TSE1
ΔT
TSE2
ΔT
TSE3
E1
E2
E3
E4
E5
E6
E7
E8
E9
MODU#1
MODU#2
MODU#3
MODU#4
MODU#5
MODU#6
MODU#7
MODU#8
MODU#9
DATA
OFFSET AMOUNT
(BYTE)
OF3
OF5
OF9
FN1
FN2
FN3
E3
E5
E9

Fig.18

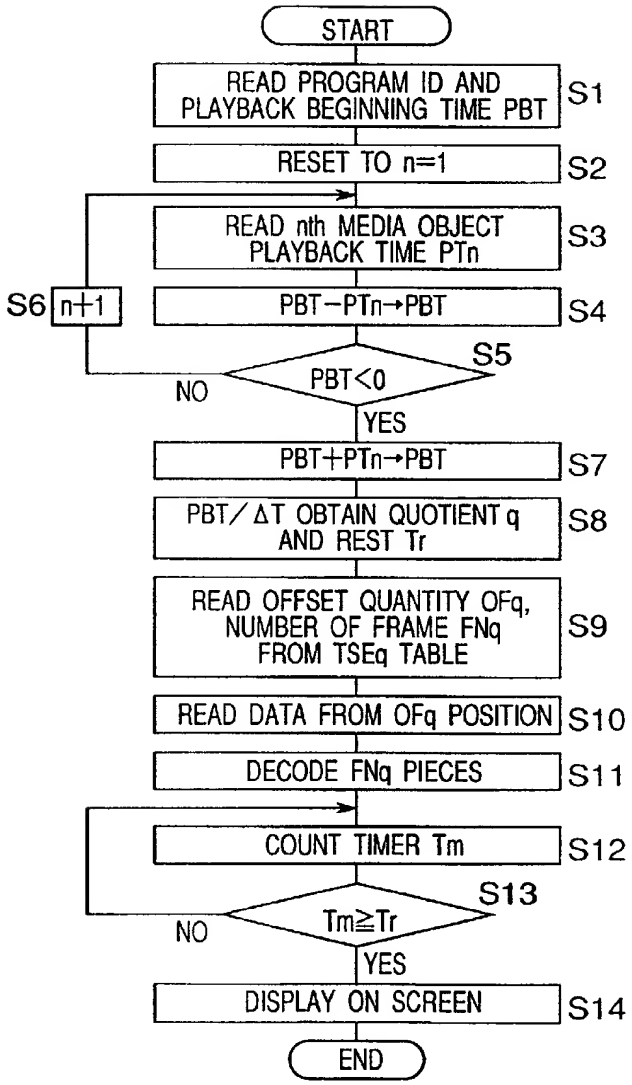

START
READ PROGRAM ID AND PLAYBACK BEGINNING TIME PBT
S1
RESET TO n=1
S2
READ nth MEDIA OBJECT PLAYBACK TIME PTn
S3
PBT−PTn→PBT
S4
S6 n+1
PBT<0
S5
NO
YES
PBT+PTn→PBT
S7
PBT/ΔT OBTAIN QUOTIENT q AND REST Tr
S8
READ OFFSET QUANTITY OFq, NUMBER OF FRAME FNq FROM TSEq TABLE
S9
READ DATA FROM OFq POSITION
S10
DECODE FNq PIECES
S11
COUNT TIMER Tm
S12
Tm≧Tr
S13
NO
YES
DISPLAY ON SCREEN
S14
END

Fig.19

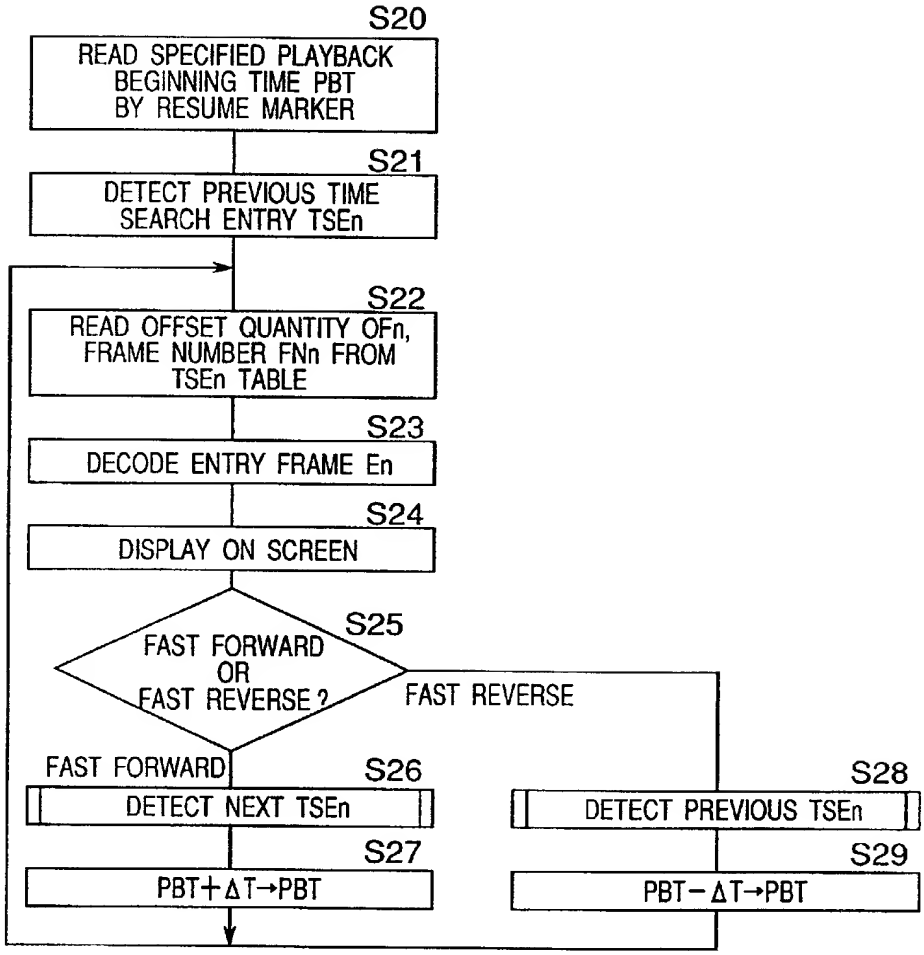

S20
READ SPECIFIED PLAYBACK BEGINNING TIME PBT BY RESUME MARKER
S21
DETECT PREVIOUS TIME SEARCH ENTRY TSEn
S22
READ OFFSET QUANTITY OFn, FRAME NUMBER FNn FROM TSEn TABLE
S23
DECODE ENTRY FRAME En
S24
DISPLAY ON SCREEN
S25
FAST FORWARD OR FAST REVERSE ?
FAST REVERSE
FAST FORWARD
S26
DETECT NEXT TSEn
S27
PBT+ΔT→PBT
S28
DETECT PREVIOUS TSEn
S29
PBT−ΔT→PBT

Fig.20

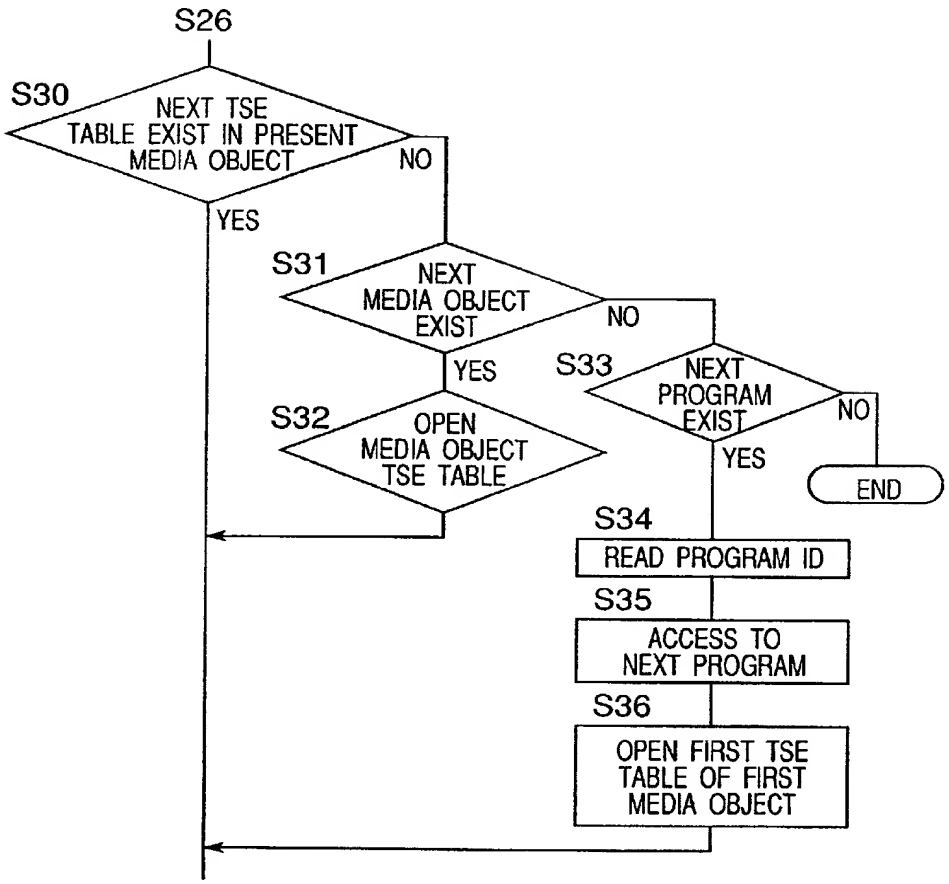

S26
S30
NEXT TSE TABLE EXIST IN PRESENT MEDIA OBJECT
NO
YES
S31
NEXT MEDIA OBJECT EXIST
NO
YES
S32
OPEN MEDIA OBJECT TSE TABLE
S33
NEXT PROGRAM EXIST
NO
YES
END
S34
READ PROGRAM ID
S35
ACCESS TO NEXT PROGRAM
S36
OPEN FIRST TSE TABLE OF FIRST MEDIA OBJECT

Fig.21

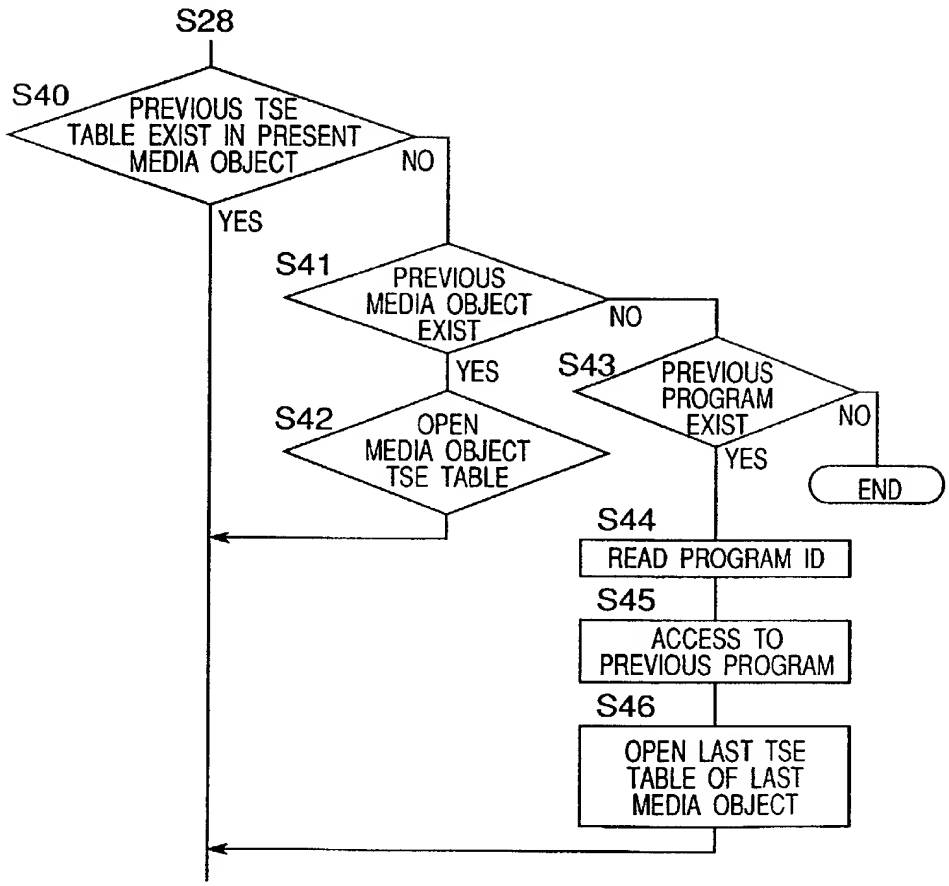
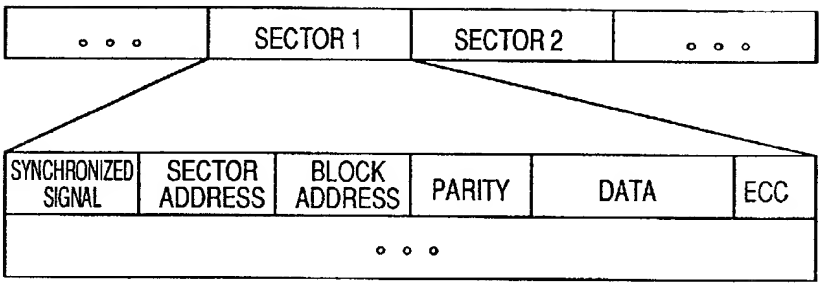

S28
S40
PREVIOUS TSE TABLE EXIST IN PRESENT MEDIA OBJECT
NO
YES
S41
PREVIOUS MEDIA OBJECT EXIST
NO
YES
S43
PREVIOUS PROGRAM EXIST
NO
YES
END
S42
OPEN MEDIA OBJECT TSE TABLE
S44
READ PROGRAM ID
S45
ACCESS TO PREVIOUS PROGRAM
S46
OPEN LAST TSE TABLE OF LAST MEDIA OBJECT

Fig.22

(a)



(b)

INDEX NUMBER	SECTOR ADDRESS
1	00000
2	0001F
3	00027

(c)

SECTOR ADDRESS	TIME CODE
00000	00 : 00 : 00
00001	00 : 00 : 01
00002	00 : 00 : 05

(d)

SECTOR ADDRESS	CONTENT
00000	A
00001	B
00002	C

(e)

SEQUENCE HEADER	SECTOR ADDRESS
SH1	00000
SH2	0001F
SH3	00027

(f)

I PICTURE	SECTOR ADDRESS
I1	00000
I2	0001F
I3	00027